**CMPG-767 Image processing and Analysis**

**Project 3**

1. Design the following functions:

   a) (**30 points)** A function for simulation of impulse noise (random impulse noise with the [0, 255] range and salt-and-pepper impulse noise), which should be used to artificially corrupt randomly selected pixels of an image). The function shall accept a clear image, a type of noise (random or salt-and-pepper), and a corruption rate as parameters and return a corrupted image.

   b) (**40 points**)

   **Graduate students**
   (undergraduate students can do this part and get *60 points*)
   A function utilizing **differential rank impulse detector** followed by **median filtering only of those pixels**, which were detected as noisy. Do not forget to take care of boundary effect and to extend an image over its boundaries using mirroring.
   The function shall accept an image, length of a rank interval ***r***, and threshold ***s*** (see slides 15-24 of Lecture-6) as parameters and return a filtered image.

   **Undergraduate students**
   A function utilizing **median filtering** without noise detection. The function shall accept a noisy image as a parameter and return a filtered image. Do not forget to take care of boundary effect and to extend an image over its boundaries using mirroring.

2. (**30 points**) Choose an image $f(x, y)$

   **a)** Generate random impulse noise $\eta(x, y)$ with the corruption rates 0.05 and 0.1 using the function, which you designed, corrupt your image $f(x, y)$, and save noisy pictures

   **b)** Generate salt and pepper noise with the corruption rate 0.05 and 0.1 using the function, which you designed, corrupt your image $f(x, y)$, and save noisy pictures

   **c)** Filter each of your noisy images using the function, which you designed in 1b).
   If you use filtering with the differential rank impulse detector, you may apply filtering with the detector more than 1 time, if necessary. Try different values of interval r, and s and find the values giving the best filtering result in terms of RMSE/PSNR.

   **d)** Find RMSE/PSNR for your filtered images (use functions designed in Project 2).

   **e)** Prepare a brief technical report containing a table of your RMSE/PSNR values or putting them as captions under respective filtered images.

3. Turn in your source code, resulting images and the report.